SUBSTITUTE SPECIFICATION

SECURED ACCESS DEVICE WITH CHIP CARD APPLICATIONS

Field of the Invention

The present invention relates to a secured access device for chip card applications. More specifically, the invention relates to a device for secured access to chip card applications that uses instructions that have been performed in the chip card which, at each instant, provide information on rights for accessing the memory of the chip card, the software component, or the hardware operation that has been performed in the chip card.

Background of the Invention

The most common type of chip card has a microprocessor that manages a program memory. The program memory is usually dedicated to a single application or a set of applications loaded at the same time into the chip card. When several applications are loaded into a chip card, they have a close relationship with one another, and are all designed for the same type of service. Thus, for example, a chip card cannot simultaneously play the role of a bank card and that of a customer card for another type of business.

In order to end this situation where each chip card has to be limited to one type of application, new software architectures are being considered. These new software architectures are making use of the development of standardized programming languages which resolve the problems of portability, such as the programming language JAVA, for example.

Figure 1 is a simplified view of a software architecture of the chip cards that are now being developed. The architecture shown in Figure 1 includes, in particular, a first part 110 that corresponds to the software architecture and a second part 120 that corresponds to the applications part of the software architecture for the chip card 100. The system part 110 is essentially formed by a library of programs 112 for the operating system of the chip card, an interface 114 to manage the interactions with the microprocessor or the different memories of the chip card, and a space for the management of hardware interruptions 116.

The applications part 120 of the software architecture includes different applications, such as a first, second and third main application, respectively 122, 124 and 126, and a first, second and third additional application, respectively 121, 123 and 125. The main applications 122, 124 and 126 are written in a programming language that can be directly understood by the processor of the chip card.

The additional applications 121, 123 and 125 are typically applications encoded in a standardized language. These applications may be added at any point in time to the system part 110. In Figure 1, the additional applications 121, 123 and 125 depend directly on the first main application 122. The first main application 122 herein serves as an interpreter

between the additional applications and the operating system by converting the codes of the additional applications into a machine language that can be understood by the programs of the operating system 112.

The software architecture that has just been described is more complex than the one currently existing in chip cards in circulation. The architecture described assumes that it is possible to add applications in a standardized programming language, possibly after the chip card is put into circulation. It is therefore more complicated to achieve a satisfactory level of security compared to when a single application or a group of applications dedicated to a single chip card function are the only applications to be loaded into the chip card. The chip card was then permanently limited in terms of available applications. The risk that a new application might disturb the operation of previous applications was therefore not as great.

The coexistence of applications of different kinds in the same chip card may raise a certain number of problems. For example, a software architecture simultaneously containing an application dedicated to the assessment of a customer's access to a gasoline company and a standard banking application must ensure that a secret key used in the banking application cannot be read during the use of the application associated with the gasoline company.

Summary of the Invention

It is an object of the present invention to overcome the problems that have just been described.

A device is provided that enables the management of different software applications that are installed, possibly at different times, or the

management of different hardware events of a chip card while providing high security. Thus, the device according to the invention offers the possibility of detection when the user of an application tries to exceed his rights, for example, by attempting to access data not intended for the application in question.

To achieve this objective, the device sets up specific instructions internal to the microprocessor of the chip card. These specific instructions are call instructions and return instructions. These call and return instructions are associated with specific registers for determining whether the operations performed by the application are authorized.

The invention therefore pertains to a device for accessing applications of a chip card comprising a microprocessor associated with an operating system working with a set of instructions, a program memory, and one or more applications in a memory of the chip card.

The device comprises a register of the microprocessor to store a code on several check bits proper to an entity brought into play. Also included are a call instruction, and an instruction for the return of the set of instructions to instantaneously and automatically update the register during the action by a new entity. The device further includes a checking device for checking, as a function of the check bits, whether access to the zones or address location of the memory of the chip card by the new entity that is called or comes into action in the chip card is authorized. A first link transmits the check bits from the microprocessor to the checking device.

According to a particular embodiment of the device of the invention, each new entity being executed is activated at a predefined address of a read only

memory (ROM) of the chip card. According to different embodiments of the invention, the entity operating in the chip card may be an application of the one or more applications or a hardware event, or the operating system associated with the microprocessor of the chip card.

Brief Description of the Drawings

The various aspects and advantages of the invention shall appear more clearly hereinafter in the following description made with reference to the appended figures which are given purely by way of an indication and in no way restrict the scope of the invention, and which are now introduced:

FIG. 1 is a simplified block diagram of a software architecture for the chip cards currently being developed according to the prior art; and

Figure 2 is a block diagram illustrating the principle of operation for the execution of an application within a chip card according to the present invention. A microprocessor 200 manages the set of operations for a plurality of applications 210 of the chip card 100.

Detailed Description of the Preferred Embodiments

A two-way bus 250 exchanges information between the microprocessor 200 and any application of the plurality of applications 210-212. The information exchanged may be data elements, addresses or control instructions. An access controller to the memory 220 exchanges information with the microprocessor 200 using a link 230, which conveys a control signal between the microprocessor 200 and the controller providing access to the memory 220.

When an entity such as the application 211, for example, requires the intervention of another entity, such as an application 212, it sends a call instruction DCALL using the two-way bus 250 followed by a designation of the entity called and a parameter enabling the nature of the call to be determined. According to the invention, a register R is updated during such calls. A certain number of bits of the register R then assume a value associated with the called entity. The register R is therefore a hardware component of the microprocessor 200 used to store a code proper to the entity of the software architecture that is being performed, and to control its field of execution.

Furthermore, the device according to the invention may also take into account instructions known as hardware instructions, such as resetting type instructions, for example. Instructions known as hardware instructions are events that may occur in real time and generate interruptions in the microprocessor of the chip card. This type of event is managed by the device in the same way as the software instructions. The bits of the register R take a very precise value appropriate to each real-time event affecting the chip card, thus limiting and controlling the rights pertaining to these events.

The information given by the register R is thus capable of checking information on the identification of the zone of the software architecture concerned by the application being executed. This information is checked at the microprocessor or at any other entity external to the software architecture.

The information given by the register R enables the checking of the zone of the memory of the chip card in which the application is permitted to be

accessed. Thus, any user attempting to make fraudulent use of the operating system in order to recover data pertaining to a particular application is refused access to this data. The bits of the state register in this case are different from the bits that might correspond to a call instruction DCALL of the particular application in question.

The addresses to be accessed and the bits of the register R sent by the microprocessor via link 230 are compared with each other in the access controller of the memory 220. If the addresses of the memory to be accessed are not addresses belonging to the authorized field of the last application having performed a call instruction DCALL, then information on illegal access to the memory is prohibited.

The device according to the invention thus provides great security in the sense that data elements intended for one application cannot be used by another application. A second register CS makes it possible to retain in memory a code proper to the applications that were active at the last call instruction DCALL sent by the current application, namely those that are to be performed following the current application.

When the current application has completed execution, a return instruction DRET is executed by the microprocessor and the data elements contained in the second register CS enable a return to the application that was being performed previously and had been activated by a call instruction DCALL. The register R is also updated.

The second register CS cannot be directly accessed by the applications of the chip card. This is to ensure the integrity of the device when it is put into operation during the execution of a return instruction DRET. When the execution of the current

application is finished, the bits of the register R assume a value specific to the application that was being performed previously, restoring its rights and limits in terms of memory access. The memory zone access device according to the invention gives a high level of security in terms of access to the different zones of the memory for a software architecture such as the one shown in Figure 1.